

A Tool to Enhance Cooperation and Knowledge Transfer among Software Developers

Seçil Aydın¹ and Deepti Mishra²

¹ Department of Computer Technology and Information Systems
Bilkent University, Ankara, Turkey
asecil@bilkent.edu.tr

² Department of Computer Engineering
Atılım University, Ankara, Turkey
deepti@atilim.edu.tr

Abstract. Software developers have been successfully tailoring software development methods according to the project situation and more so in small scale software development organizations. There is a need to share this knowledge with other developers who may be facing the same project situation so that they can benefit from other people experiences. In this paper, an approach to enhance cooperation among software developers, in terms of sharing the knowledge that was used successfully in past projects, is proposed. A web-based tool is developed that can assist in creation, storage and extraction of methods related with requirement elicitation phase. These methods are categorized according to certain criteria which helps in searching a method that will be most appropriate in a given project situation. This approach and tool can also be used for other software development activities.

Keywords: Method, Software development, Cooperation, Knowledge Transfer.

1 Introduction

Software systems are getting increasingly complicated today; the knowledge needed for the implementation is vast and unlikely to be held by any individual software developers [7]. Methods and techniques for software development are normally general in nature and they can not be used directly without adapting them according to the characteristics of the project. It is also important for the software engineers to learn from their own past experience as well as experience of others. Therefore, there is a need to enhance the cooperation among software developers and software development organizations must have a mechanism to support the sharing of knowledge held by different software developers to remain competitive.

In this paper, an approach to enhance cooperation and knowledge sharing among software developers is proposed with the help of a web-based tool. Methods that are used successfully in past projects, along with project characteristics, can be stored and shared with the help of this tool. This tool also provides support for the reuse of existing methods to create new methods.

This paper is organized as follows: In next section, criteria to classify different requirement elicitation techniques are established. In section 3, use of the tool for storing and sharing the methods is explained. Finally, the paper concludes in section 4.

2 Criteria for Classifying Requirement Elicitation Techniques

IEEE [5] defines requirements engineering as the process of studying user needs to arrive at a definition of whole system requirements. There are variety of techniques that can be used to understand user and stakeholder requirements. Because of the relative strengths and weaknesses of the available approaches, most projects require a combination of several techniques in order to produce quality results [6]. Clearly requirement elicitation does not occur in a vacuum and is highly dependent on some characteristics [3]. We studied different requirement elicitation techniques and their suitability in different project situations. With this knowledge, following criteria are formed to classify different requirement elicitation techniques.

- Experience of the requirement engineer
- Requirement elicitation period
- Experience level of the company's customer
- Possibility of meeting between development team and customers
- Project budget
- Project's user interaction level
- Project Complexity

3 Tool Description

A new, user friendly and progressive web-based tool for requirement engineering phase is designed to demonstrate the proposed solution. We preferred to develop a web-based tool because the project members can be in different physical locations still they can easily access it.

There are two types of user in the system as shown in Figure 1: admin and user. Admin can perform all the action which a normal user can perform. Additionally, admin can add/delete user(s) in the system, allocate password(s) to the users, delete a method.

Methods that are stored can be viewed by View Method module. User can search methods based on the criteria (explained in section 2) to best suit the project situation at hand. If the desired method is not found, our system enables user to create a new method. There are three ways to create a new method. A method can be created by an empty method template, by modifying a method stored in the repository and by combining two methods stored in the repository. This type of storage provides flexibility to the user to emphasize and transfer his/her experience. Methods are considered as a single, whole unit for reuse purpose because it is important for the user to see complete method so that he/she can understand it and if require can create a new method by updating an existing method or by assembling two methods.

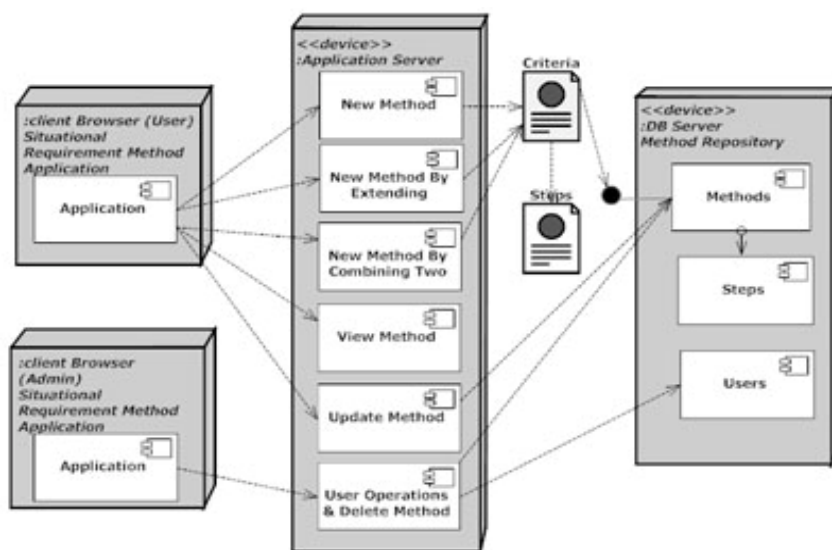


Fig. 1. Structure of the Tool

If user wants to add new method, it can be done by using “Add New Method” functionality. Method name is entered and the criteria related with the method are selected from the form then Add Step page is displayed to enter the steps of the method.

If user wants to add new method by extending an existing method stored in the system, “Add New Method By Extending One” functionality is used. It first lists all methods stored in the system then user selects the method and later the method information is listed. User is required to enter the new method name and asked to change the criteria for the new method (if required) and then change/add new steps.

If user wants to add new method by combining two methods stored in the system, “Add New Method By Combining Two” functionality is used. It first lists all methods stored in the system then user selects any two methods. After that the methods information are listed. User enters the new method name and selects the criteria and then selects the steps he/she wants to add from existing two methods he/she has chosen.

If user wants to update an existing method, Update Method module is used. All of the methods available in the system are listed in Update Method page. User selects the method he/she wants to update then all related information of the selected method is listed. First the criteria are updated then the steps are updated. Only the author of a method can update the method using the tool.

Although there exist several studies in the literature which explores situational method engineering but there are very few academic tool prototypes and tools to use for this issue. One of the tools for situational method engineering is Decamerone [2]. It has own language and interpreter so the user should learn the language of that tool to be able to use it. Another technique is the Noesis technique that allows recursive and compositional structures to be captured in the meta-models and situational

methods to be obtained by the assembly of method fragments [4]. This technique also requires more expertise to use. Our tool is more easy to use and flexible. Middle level expertise is enough for the usage.

James Martin & Co.'s CAME tool which supports many of seven features of Harmsen's [1] offers a choice of project objectives. Based on these, tool can generate an appropriate methodology. Despite this, our tool stores the methods as a whole so influence of user on the selected method is more than the other tools. Moreover, CMMI also supports storing lessons learned databases which can be facilitated by using our proposed tool.

4 Conclusion

It has been found that software developers are successfully tailoring standard software development methods according to their project situations. The main problem is that this knowledge and experience is not communicated to other people who might be facing the same project situations especially in small software development organizations. A simple and formal approach to method tailoring, and storage can solve this problem and developers can learn from each others experiences. This will also enhance their productivity. We have developed a simple approach for method storage, sharing and retrieval according to project characteristics with the help of a web-based tool. New methods can be developed from scratch, by extension or assembling existing methods. We applied this approach and tool specifically for requirement elicitation methods but this approach and tool can be applied for any software development phase.

References

1. Brinkkemper, S., Lyytinen, K., Welke, R.J.: *Method Engineering: Principles of Method Construction and Tool Support*. Chapman & Hall, Boca Raton (1996)
2. Brinkkemper, S., Harmsen, F.: *Design and Implementation of a Method Base Management System for a Situational CASE Environment*. IEEE, Los Alamitos (1995)
3. Christel, M.G., Kang, K.C.: *Issues in requirements elicitation*, Software Engineering Institute, Carnegie Mellon University, pittsburg, PA, Carnegie Mellon University Technical Report. CMU/SEI-92-TR-012 (1992)
4. Domínguez, E., Zapata, M.A.: Noesis: Towards a situational method engineering technique. *Inf. Syst.* 32(2), 181–222 (2007)
5. IEEE Standard Glossary of Software Engineering Terminology, IEEE (1990)
6. Maiden, R., Rugg, G.: ACRE: selecting methods for requirements acquisition. *Software Engineering Journal* 11(3), 183–192 (1996)
7. Ye, Y., Yamanoto, Y., Kishida, K.: *Dynamic Community: A New Conceptual Framework for Supporting Knowledge Collaboration in Software Development*. In: *Proceedings of the 11th Asia-Pacific Software Engineering Conference* (2004)